LABORATORIO 2

Unidad 3

Camila Paladines

Computación Científica

Profesor: Hernán Darío Vargas Cardona, PhD

Abril 9 de 2021

RESUMEN

En este informe se describen los resultados conseguidos al implementar algoritmos en Python para la obtención de respuestas a problemas matemáticos, como lo es el cómputo de los parámetros que proporcionan un ajuste lineal para un conjunto de datos, en este caso, mediante el Método de Ecuaciones Normales y el Método de Transformaciones Householder. Además, se realizan algunos análisis sobre los métodos utilizados aplicándolos a conjuntos de datos reales, donde se observa su comportamiento dependiendo de diferentes factores. También se calcula la complejidad computacional y la exactitud, con el fin de comparar los métodos y determinar cuál obtuvo mejores resultados para los conjuntos de datos usados.

ABSTRACT

This report describes the results obtained by implementing algorithms in Python to obtain answers to mathematical problems, such as the computation of the parameters that provide a linear fit to a data set, in this case, using the Normal Equations Method and the Householder Transformations Method. In addition, some analyses are performed on the methods used by applying them to real data sets, where their behavior is observed depending on different factors. The computational complexity and accuracy are also calculated, in order to compare the methods and determine which one obtained better results for the data sets used.

Contenido

1. Introducción

En el mundo físico que conocemos existen algunos elementos o comportamientos que siguen ciertos patrones. Algunos de ellos son, por ejemplo, la sucesión de Fibonacci, presente en el crecimiento de la población de conejos, la cantidad de pétalos en algunas flores, etc. Desde el punto de vista matemático, se puede *predecir* el valor de salida para un valor de entrada (o un conjunto de valores), dado un patrón que se haya encontrado en ellos. Para realizar esta tarea se han planteado diferentes métodos, en este caso, se usarán el Método de Ecuaciones Normales y el Método de Transformaciones Householder para obtener los parámetros de una función y usarla para predecir los datos de salida dependiendo de un valor de entrada.

Estas herramientas se pueden aplicar a diferentes campos, siendo el aprendizaje automático el más destacado entre ellos, donde uno de sus objetivos principales es entrenar un modelo de predicción con un conjunto de entrenamiento y probar su exactitud con otro conjunto de validación que permita verificar que este modelo realmente aprendió.

2. Materiales y métodos

2.1. Materiales

Para el desarrollo de esta unidad se usó Python 3.7, con las librerías `numpy`, `matplotlib.pyplot`, `time` y `pandas`.

2.2. Métodos

2.2.1. Ajuste de Datos

Dados m datos (t_i, y_i), se busca encontrar el vector x de parámetros que provea el mejor ajuste a una función $f(t, x)$, donde $f : \mathbb{R}^{n+1} \to \mathbb{R}$. El ajuste está dado por:

$$min\, x \sum_{i=1}^{m} (y_i - f(t_i, x))^2$$

que se conoce como la solución de mínimos cuadrados, la cual es lineal si la función f es lineal en los componentes del vector x. Esto significa que f es una combinación lineal de funciones ϕ que dependen de t:

$$f(t, x) = x_1\phi_1(t) + x_2\phi_2(t) + \cdots + x_n\phi_n(t)$$

Un ajuste lineal polinómico se puede expresar de la siguiente manera:

$$f(t, x) = x_1 + x_2 t + x_3 t^2 + \cdots + x_n t^{n-1}$$

2.2.2. Mínimos Cuadrados Lineal

Un problema de mínimos cuadrados lineal se puede escribir en notación matricial:

$$Ax \approx b$$

Donde $A_{ij} = \phi_j(t_i)$ y $b_i = y_i$. Por ejemplo, en el ajuste polinomial cuadrático el cual tiene tres parámetros, y teniendo un conjunto de 5 datos (t_1,y_1), ...,

(t_5,y_5), entonces la matriz A sería de 5×3, y el problema tiene la forma:

$$Ax = \begin{bmatrix} 1 & t_1 & t_1^2 \\ 1 & t_2 & t_1^2 \\ 1 & t_3 & t_1^3 \\ 1 & t_4 & t_1^4 \\ 1 & t_5 & t_1^5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \approx \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} = b$$

2.2.3. Método de Ecuaciones Normales

Un sistema de ecuaciones normales de $n \times n$ dado por $A^\top Ax = A^\top b$ se puede usar para solucionar el problema de mínimos cuadrados lineal $Ax \approx b$.

La matriz $A^\top A$ es simétrica y positiva definida, por lo cual se puede computar su descomposición de Cholesky:

$$A^\top A = LL^\top$$

donde L es una matriz triangular inferior y $L^\top$ triangular superior. La solución x del problema de mínimos cuadrados se puede computar solucionando dos sistemas triangulares: $Ly = A^\top b$ y $L^\top x = y$.

2.2.4. Método de Transformaciones Householder

Mediante transformaciones Householder sucesivas, se puede introducir ceros columna por columna, debajo de la diagonal de una matriz A, para llevarla a la forma triangular. Cuando se aplica esta transformación a un vector x, se nota que:

$$Hx = \left(I - 2\frac{vv^\top}{v^\top v}\right) x = x - 2\left(\frac{v^\top x}{v^\top v}\right) v$$

siendo la expresión de la derecha más barata computacionalmente y sólo requiere conocer el vector v. Este proceso sucesivo produce una factorización de la forma:

$$H_n \dots H_1 A = \begin{bmatrix} R \\ O \end{bmatrix}$$

donde R es triangular superior.

El producto de transformaciones Householder sucesivas $H_n \dots H_1$ es también una matriz ortogonal. Por lo tanto, si se toma:

$$Q^\top = H_n \dots H_1,\ Q = H_1^\top \dots H_n^\top$$

Entonces:

$$A = Q \begin{bmatrix} R \\ O \end{bmatrix}$$

Esto quiere decir que al final se debe resolver el siguiente problema de mínimos cuadrados triangular equivalente:

$$\begin{bmatrix} R \\ O \end{bmatrix} x \approx Q^\top b$$

3. Resultados de las simulaciones

Los datos para la realización de las pruebas fueron tomados de GitHub. Allí se almacenan las estadísticas de los contagiados, muertos y recuperados por COVID-19. Como el repositorio se actualiza diariamente, se tomaron los datos desde el 22 de enero de 2020 hasta el 1 de abril de 2021.

3.1. Ejemplo 1

Este conjunto de datos describe la cantidad de muertos al día por COVID-19, para este laboratorio sólo se tomaron los 50 más recientes.

3.1.1. Parámetros x

Los parámetros calculados (con $n = 2, \ldots, 6$) por ambos métodos son los que se muestran en la siguiente tabla:

n	x_1	x_2	x_3	x_4	x_5	x_6
2	2359906.63	9002.15				
3	2357477.49	9293.41	-5.82			
4	2351274.95	10775.12	-79.84	0.99		
5	2352260.35	10386.68	-45.02	-0.09	0.01	
6	2352351.72	10333.72	-37.69	-0.48	0.02	0

Aunque el Método de Transformaciones Householder trabaja muy bien hasta $n = 20$, para el alcance de este laboratorio se mantuvo hasta $n = 6$, que es hasta donde el Método de Ecuaciones Normales no tiene problemas.

3.1.2. Tiempo de cómputo

El tiempo de ejecución (en segundos) para cada método según n se muestra en la siguiente tabla:

n	Ecuaciones Normales	Transformaciones Householder
2	0.000277	0.000436
3	0.000259	0.000664
4	0.000263	0.000982
5	0.000311	0.001384
6	0.001242	0.003573

Que se puede observar mejor en la siguiente gráfica, donde la línea roja representa el tiempo que gastó el Método de Ecuaciones Normales, y la línea azul el Método de Transformaciones Householder:

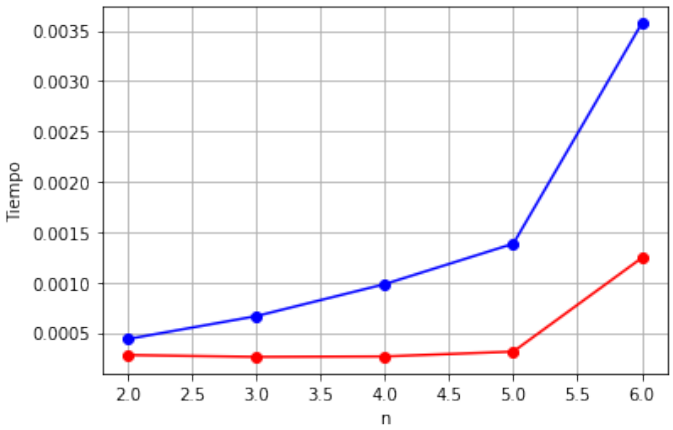

Figura 1: n vs tiempo

3.1.3. Exactitud de los algoritmos

La exactitud de los algoritmos se obtuvo mediante el Error Cuadrático Medio. En la siguiente tabla se encuentra el correspondiente ECM para cada n:

n	Ecuaciones Normales	Transformaciones Householder
2	10525755.72	10525755.72
3	10883309.88	10883309.88
4	4687698.73	4687698.73
5	4089110.24	4089110.24
6	4042539.55	4042539.55

Que se puede observar mejor en la siguiente gráfica (como ambos métodos tienen los mismos valores sólo se puede ver una de ellas):

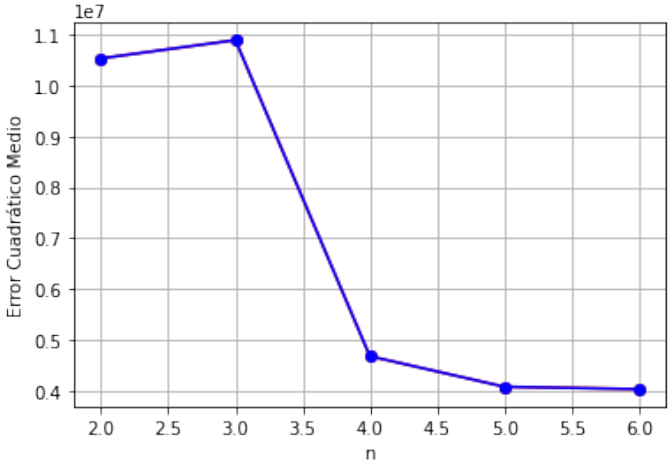

Figura 2: n vs ECM

3.1.4. Gráfica del ajuste de mínimos cuadrados

Como se observó en la sección anterior, el mejor valor de n (con menor ECM) es 6, por lo que en la siguiente gráfica (tomando los parámetros para $n = 6$) se muestra los datos de entrenamiento (azul) y validación (rojo), además de la gráfica de la función con los parámetros calculados.

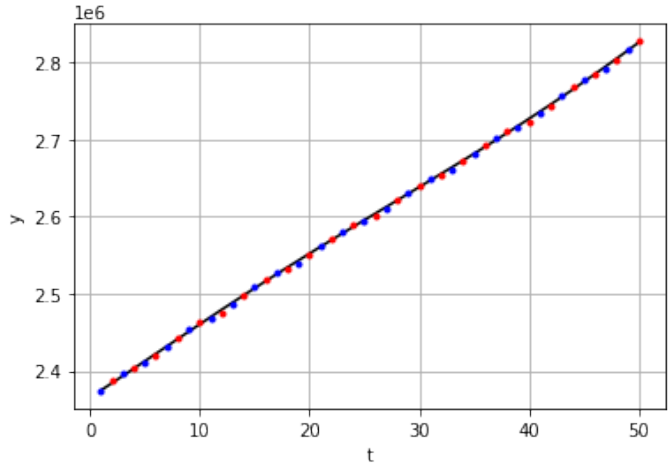

Figura 3: Ajuste de datos (t vs y)

3.2. Ejemplo 2

Este conjunto de datos describe la cantidad de recuperados al día por COVID-19, para este laboratorio sólo se tomaron los 50 más recientes.

3.2.1. Parámetros x

Los parámetros calculados (con $n = 2, \ldots, 6$) por ambos métodos son los que se muestran en la siguiente tabla:

n	x_1	x_2	x_3	x_4	x_5	x_6
2	59882434.57	258349.20				
3	60331351.06	204522.28	1076.54			
4	60089679.61	262254.26	-1807.37	38.45		
5	60063479.05	272582.33	-2733.15	67.22	-0.29	
6	60040907.29	285667.29	-4545.33	163.47	-2.45	0.02

Aunque el Método de Transformaciones Householder trabaja muy bien hasta $n = 20$, para el alcance de este laboratorio se mantuvo hasta $n = 6$, que es hasta donde el Método de Ecuaciones Normales no tiene problemas.

3.2.2. Tiempo de cómputo

El tiempo de ejecución (en segundos) para cada método según n se muestra en la siguiente tabla:

n	Ecuaciones Normales	Transformaciones Householder
2	0.000274	0.0004398
3	0.000236	0.0006533
4	0.000267	0.0010018
5	0.000314	0.0014212
6	0.000557	0.0034406

Que se puede observar mejor en la siguiente gráfica, donde la línea roja representa el tiempo que gastó el Método de Ecuaciones Normales, y la línea azul el Método de Transformaciones Householder:

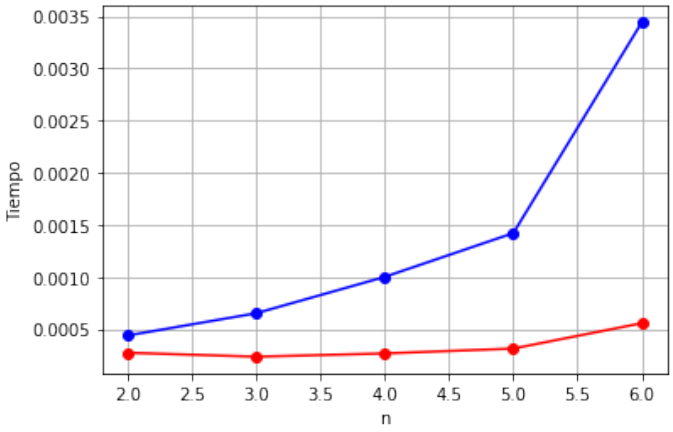

Figura 4: n vs tiempo

3.2.3. Exactitud de los algoritmos

La exactitud de los algoritmos se obtuvo mediante el Error Cuadrático Medio. En la siguiente tabla se encuentra el correspondiente ECM para cada n:

n	Ecuaciones Normales	Transformaciones Householder
2	54142465742.89	54142465742.89
3	7436148459.31	7436148459.31
4	1014665320.75	1014665320.75
5	811714625.91	811714625.91
6	1009299499.68	1009299499.68

Que se puede observar mejor en la siguiente gráfica (como ambos métodos tienen los mismos valores sólo se puede ver una de ellas):

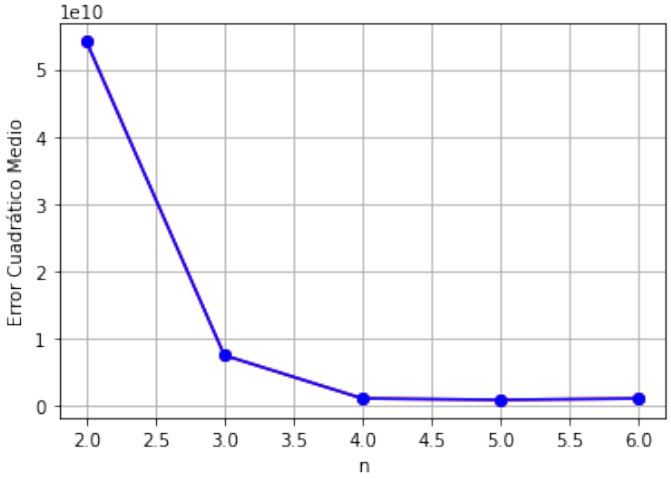

Figura 5: n vs ECM

3.2.4. Gráfica del ajuste de mínimos cuadrados

Como se observó en la sección anterior, el mejor valor de n (con menor ECM) es 5, por lo que en la siguiente gráfica (tomando los parámetros para $n = 5$) se muestra los datos de entrenamiento (azul) y validación (rojo), además de la gráfica de la función con los parámetros calculados.

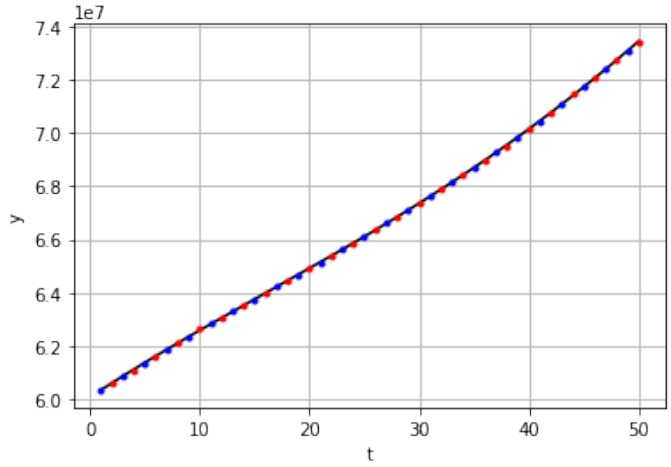

Figura 6: Ajuste de datos (t vs y)

4. Discusión y análisis

Al usar los métodos implementados en conjuntos de datos reales, se pueden establecer las siguientes consideraciones.

4.1. Resultados de los métodos

Tanto para el ejemplo 1 como para el ejemplo 2, los resultados de ambos métodos fueron los mismos (en estos conjuntos de datos específicos), lo que implica que se puede elegir cualquiera de los dos dependiendo de las necesidades, es decir, que si se quiere resultados para $n \leq 6$ se debería usar el Método de Ecuaciones Normales ya que es más rápido, pero si se quiere obtener resultados para $n > 6$ se debe usar el Método de Transformaciones Householder aunque se demore n veces más.

4.2. Límite de los algoritmos

Para ambos ejemplos, el n más alto hasta donde el Método de Ecuaciones Normales logró dar la respuesta es $n = 6$, ya que para $n \geq 7$ ocurría un error en la función `cholesky()` tomada de Numpy. En el caso del Método de Transformaciones Householder el límite fue $n = 125$ por los alcances que tiene el lenguaje en el que se implementaron los algoritmos. Como ya se había mencionado, se realizaron los análisis para $n = 2, \ldots, 6$ para poder comparar los métodos en cada ejemplo.

4.3. Complejidad computacional

La complejidad computacional del Método de Transformaciones Householder es más alta que la del Método de Ecuaciones Normales, ya que la primera debe hacer varios cálculos que implican ciclos anidados de $n \times n \times m$, mientras que en la segunda se aproxima una complejidad de $n \times m$. Esto quiere decir que a medida que crece n la distancia entre los tiempos de ejecución de los métodos

se vuelve más amplia, como se puede observar en las gráficas de la Figura 1 y la Figura 4.

4.4. Mejores valores de n

En el ejemplo 1, el valor de n para el cual se obtuvo el mejor ajuste polinomial fue $n = 6$, mientras que en el ejemplo 2 fue $n = 5$. En el caso del Método de Transformaciones Householder, como su límite es $n = 125$ sería posible establecer el mejor valor para cada uno de los ejemplos, pero sería necesario aumentar el valor de m (cantidad de datos) y compararlos de nuevo. En este sentido, con $m = 150$ y $n \leq 125$ el mejor valor de n para el ejemplo 1 es $n = 6$ con un Error Cuadrático Medio $ECM = 372707650$, y para el ejemplo 2 es $n = 6$ con un Error Cuadrático Medio $ECM = 528850405634$.

5. Conclusiones

Se logró comprender las diferentes formas de obtener los parámetros de una función para un ajuste polinomial y la importancia de los métodos aprendidos para diversas aplicaciones.

Al usar la computación para resolver problemas matemáticos se puede profundizar en el comportamiento de los métodos tanto a nivel de exactitud como a nivel de complejidad, con lo que se logra aprender más sobre ellos e identificar en qué caso usarlos y de qué manera se puede aprovechar mejor.

Es importante experimentar con las diferentes formas de solucionar un problema matemático, en especial si se implementa en un computador, pues permite obtener resultados de una manera ágil una vez se haya construido, con lo que se facilita el uso del método cuando ya se ha comprendido o cuando comprenderlo no es el objetivo principal.

6. Referencias

- Material del curso, disponible en BlackBoard
- Bornemann, F., 2016. *Numerical linear algebra*. 1st ed. *Simson, W.*
- Mathews, J., Fink, K., Fernández Carrión, A. & Contreras Márquez, M., 2011. *Métodos Numéricos con MATLAB*. 3rd ed. Madrid: *Pearson Prentice Hall*.
- Repositorio en GitHub de los conjuntos de datos
- Librería Numpy
- Librería Pyplot (Matplotlib)
- Librería Time
- Librería Pandas
- Error Cuadrático Medio (ECM)